

**Amendments to the Claims:**

1. (Currently Amended) A method ~~for demultiplexing an incoming packet to a virtual machine (“VM”)~~, comprising:  
unmapping a guest physical address for a first virtual machine (“VM”) from a host physical address in at least one page table entry associated with buffers allocated to the first VM in a direct memory access (“DMA”) table for the first VM to create unmapped buffers;  
placing the incoming packet into at least one of the unmapped buffers; and  
allocating the at least one of the unmapped buffers to [[the]] a second VM to create a mapped buffer.
2. (Original) The method according to Claim 1 wherein unmapping the guest physical address from the host physical address further comprises clearing the contents of a physical page associated with the host physical address.
3. (Currently Amended) The method according to Claim 1 wherein allocating the at least one of the unmapped buffers further comprises temporarily assigning the at least one of the unmapped buffers to the second VM to create the mapped buffer.
4. (Currently Amended) The method according to Claim 1 further comprising:  
causing the first VM to release the mapped buffer; and  
unmapping the guest physical address from the host physical address.
5. (Currently Amended) The method according to Claim 4 wherein causing the first VM to release the mapped buffer further comprises injecting a signal into the first VM.
6. (Original) The method according to Claim 5 wherein the signal is an interrupt.

7. (Currently Amended) A method for demultiplexing an incoming packet to multiple VMs, comprising:  
decoupling a guest physical address for a first virtual machine (“VM”) from a host physical address to release buffers allocated to the first VM to create unmapped buffers;  
placing incoming packets in the unmapped buffers;  
examining the incoming packets to determine appropriate destination VMs; and  
assigning the unmapped buffers to the appropriate destination VMs, wherein the appropriate destination VMs comprise a second VM different from the first VM.
8. (Currently Amended) The method according to Claim 7 wherein decoupling the guest physical address from the host physical address further comprises invalidating entries in at least one page table entry for buffers in a direct memory access table associated with the first VM.
9. (Currently Amended) A system for demultiplexing an incoming packet to an appropriate virtual machine (“VM”), comprising;  
a plurality of VMs;  
a component coupled to the plurality of VMs, the component capable of invalidating entries in at least one page table entry for direct memory access (“DMA”) buffers allocated to a first VM of the plurality of VMs to create unmapped buffers, placing the incoming packet in the unmapped buffers, determining which of the plurality of VMs is the appropriate destination virtual machine (“VM”) for the incoming packet and assigning the unmapped buffers with the incoming packet to the appropriate destination virtual machine, wherein the appropriate destination virtual machine is a second VM of the plurality of VMs, the second VM being different from the first VM.
10. (Original) The system according to Claim 9 wherein the component is one of a demultiplexer and a virtual network interface card (“VNIC”).

11. (Original) The system according to Claim 10 wherein the VNIC is maintained by a virtual machine manager (“VMM”) coupled to the plurality of VMs.
12. (Currently Amended) An article comprising a machine-accessible storage medium having stored thereon instructions that, when executed by a machine, cause the machine to demultiplex an incoming packet to a virtual machine (“VM”) by:  
unmapping a guest physical address associated with a first VM from a host physical address in at least one page table entry for buffers allocated to the first VM in a direct memory access (“DMA”) table for the first VM to create unmapped buffers;  
placing the incoming packet into at least one of the unmapped buffers; and  
allocating the at least one of the unmapped buffers to the VM a second VM to create a mapped buffer.
13. (Original) The article according to Claim 12 wherein the instructions, when executed by the machine, further cause the machine to unmap the guest physical address from the host physical address further by clearing the contents of a physical page associated with the host physical address.
14. (Currently Amended) The article according to Claim 12 wherein the instructions, when executed by the machine, further cause the machine to allocate the at least one of the unmapped buffers by temporarily assigning the at least one of the unmapped buffers to the second VM to create the mapped buffer.
15. (Currently Amended) The article according to Claim 12 wherein the instructions, when executed by the machine, further cause the machine to demultiplex an incoming packet by:  
causing the first VM to release the mapped buffer; and  
unmapping the guest physical address from the host physical address.

16. (Currently Amended) The article according to Claim 15 wherein the instructions, when executed by the machine, further cause the first VM to release the mapped buffer by injecting a signal into the first VM.

17. (Previously Presented) The article according to Claim 16 wherein the signal is an interrupt.

18. (Currently Amended) The article according to Claim 17 wherein the instructions, when executed by the machine, further cause the first VM to release the mapped buffer by injecting an interrupt into the first VM.

19. (Currently Amended) An article comprising a machine-accessible medium having stored thereon instructions that, when executed by a machine, cause the machine to demultiplex an incoming packet to multiple VMs by:  
decoupling a guest physical address for a first virtual machine (“VM”) from a host physical address to release buffers allocated to the first VM to create unmapped buffers;  
placing incoming packets in the unmapped buffers;  
examining the incoming packets to determine appropriate destination VMs; and  
assigning the unmapped buffers to the appropriate destination VMs, wherein the appropriate destination virtual machine is a second VM different from the first VM.

20. (Currently Amended) The article according to Claim 19 wherein the instructions, when executed by the machine further decouple the guest physical address from the host physical address further by invalidating entries in a direct memory access table associated with the first VM.